entering into a secure connection with a server computer;

obtaining a session key specific to the secure connection;

receiving data encrypted with the session key from the server computer;

storing the encrypted data on a persistent storage; and

securing the session key with a storage key.

REMARKS

In view of the following remarks, Applicant respectfully requests reconsideration and allowance of the subject application. No new matter is added by any amendment. Amendments made to claims 1, 2, 8, 11, 12 and 18-22 are not made for the purpose of patentability with respect to any prior art of record, and such amendments do not affect the allowability of allowed claims 18-22.

Drawings

The Draftsperson has objected to Figures 1, 4, 5, 7A and 10 of the drawings under 37 CFR §1.84, indicating the top margins are not acceptable. The Draftsperson has also objected to Figures 2-11 of the drawings under 37 CFR §1.84, indicating that lines, numbers and letters are not uniformly thick and well defined. The Draftsperson has also objected to Figures 1B-11 of the drawings under 37 CFR §1.84, indicating that the Figure legends are poor. As indicated above, Figures 1A, 1B, 2, 3, 4, 5, 6, 7A, 7B, 7C, 8, 9, 10 and 11 are submitted herewith in a set of formal drawings. The set of formal drawings addresses the Draftsperson's objections so that Figures 1A, 1B, 2, 3, 4, 5, 6, 7A, 7B, 7C, 8, 9, 10 and 11 comply with 37 CFR §1.84.

§102 Rejections

Claims 1-2, 5, 13-14 and **23-24** are rejected under 35 U.S.C. §102(b) as allegedly being anticipated by Herbert (US # 5,757,919). Applicant respectfully traverses the rejection.

Herbert discloses a "method and system for maintaining integrity and confidentiality of pages paged to an external storage unit from a physically secure environment" (col. 1, lines 58-60). The context of Herbert's disclosure is a single computer system in which virtual memory uses external memory devices (e.g., hard disk drives or magnetic tape) to "ameliorate the physical memory constraints of the RAM and create the appearance that adequate space is available in the RAM to hold all the currently needed code and data" (col. 1, lines 17-23). Generally, virtual memory permits a computer to execute a program that is too large to fit into the main memory (i.e., RAM) all at once. The computer executes such a program by "paging" or "swapping" blocks (e.g., 4 kilobyte pages) of code or data between RAM and external memory as needed during program execution. (col. 1, lines 19-35). Thus, Herbert discloses a way of maintaining the integrity and confidentiality of pages paged between RAM in a physically secure environment and a memory device in an external insecure environment. (col. 1, lines 58-62).

Herbert discusses ways of creating a physically secure environment such as tamper-resistant packaging materials, tamper-resistant die coatings and tamper-resistant wafer coatings used to create secure semiconductor devices and other secure electronic circuitry. Herbert further describes a physically secure environment that includes a processor coupled by a bus to a RAM. Within the physically secure environment is an integrity check engine that generates a hash

value when data is paged out of the secure environment. The hash value is "stored within the secure environment as an integrity check value (ICV) for later comparison when that page of data is subsequently paged back in" (col. 2, lines 25-63).

In rejecting **claim 1**, the Office asserts among other things, that Herbert teaches a computerized method for key-based secure storage comprising downloading content and an access predicate that specifies requirements for an application to access the content. However, as made clear from the following discussion, this is not the case.

In contrast to Herbert's system for secure paging between a secure environment and an external memory, Applicant's **claim 1** recites "downloading information and an access predicate that specifies requirements for an application to access the content". Herbert does not discuss downloading information. Herbert's disclosure relates to paging pages of code or data to facilitate a virtual memory system within a computer. Herbert discloses a "method and system for maintaining integrity and confidentiality of pages paged to an external storage unit from a physically secure environment" (col. 1, lines 58-60). Herbert states that the "functioning of paging systems is generally well understood in the art" (col. 1, lines 38-40). It is therefore clear that the term "paging" as described in Herbert (col. 1, lines 13-50) and as well understood in the art of virtual memory systems does not mean "downloading" as recited in Applicant's claim 1.

Furthermore, Herbert does not discuss an "access predicate" as recited in Applicant's claim 1. The access predicate "specifies requirements for an application to access the [downloaded] information". By contrast, Herbert discusses an "integrity check value (ICV)". Herbert's ICV does not specify

requirements for an application to access downloaded information. Rather, the ICV is used to check the integrity of a page of data previously paged out of the physically secure environment that is being paged back into the physically secure environment. This provides "security from substitution and modification attacks of programs and data beyond the memory capacity of a secure environment" (Herbert, col. 1, lines 51-55). Herbert's ICV, used to check the integrity of data being paged back into a physically secure environment, is nothing at all like an "access predicate that specifies requirements for an application to access the [downloaded] information" as recited in Applicant's claim 1.

Moreover, even if Herbert's ICV could be likened to Applicant's access predicate, Herbert does not discuss downloading an access predicate. As discussed above, Herbert does not download anything. Rather, Herbert pages code or data within a computer, which is a function well known in the art of virtual memory systems. However, it is additionally noteworthy that Herbert does not even page the ICV out of the physically secure environment. The ICV does not leave the physically secure environment of Herbert. The ICV is "stored within the secure environment . . . for later comparison when [a] page of data is subsequently paged back in" (col. 2, lines 60-63). Thus, Herbert's ICV is not even paged out of the physically secure environment, let alone downloaded to another computer. Storing an ICV in the physically secure environment as described in Herbert is in no way the same as "downloading . . . an access predicate that specifies requirements for an application to access the information" as recited in Applicant's claim 1.

It is therefore clear that Herbert does not teach "downloading information and an access predicate that specifies requirements for an application to access the

information" as recited in Applicant's claim 1. For at least the reasons discussed above, claim 1 is not anticipated by Herbert, and the §102(b) rejection of claim 1 should be withdrawn.

Claims 2, 5 and **13** depend from claim 1. Because claim 1 is allowable as discussed above, claims 2, 5 and 13 are also allowable by virtue of at least their dependency from claim 1. Applicant therefore respectfully requests withdrawal of the §102(b) rejection to dependent claims 2, 5 and 13.

Furthermore, with respect to claim 2, the Office states that Herbert teaches (at col. 7, lines 5-22) decrypting the content for access by an application only if the application meets the requirements specified in the access predicate. The Office again implies that Herbert's ICV is the same as Applicant's access predicate, which as clarified above, is not the case. Assuming for the sake of discussion, however, that Herbert's ICV is the same as Applicant's access predicate, Herbert still does not teach "decrypting the information for access by an application only if the application meets the requirements specified in the access predicate" as the Office asserts. In Herbert, the incoming page is already decrypted before the ICV's are compared. In fact, the incoming page is decrypted and then hashed to determine the ICV (col. 7, lines 12-18). Thus, the ICV does not provide requirements that determine whether the page will or will not be decrypted. Likewise, the "key" and "IV" of Herbert do not provide requirements that determine whether the page will or will not be decrypted. For these additional reasons, claim 2 is not anticipated by Herbert, and the §102(b) rejection of claim 2 should be withdrawn.

In rejecting **claim 14**, the Office asserts that Herbert teaches (at col. 3, lines 2-8 and col. 2, lines 39-45) a generate key function executed from a computer-

readable medium by a processing unit, wherein the generate key function causes the processing unit to generate an operating system storage key based on an identity for the operating system. However, Herbert's Figure 1 and corresponding discussion at col. 2, line 39 through col. 3, line 5 make clear that this is not an accurate interpretation of Herbert's teaching.

Herbert specifies a "random number generator 18 . . . to generate keying material for the encryption engine 12". Although Herbert teaches a physically secure environment 1 that contains a processor 16, Herbert does not teach "a generate key function executed from [a] computer-readable medium by the processing unit". The random number generator 18 of Herbert is not a generate key function executed from a computer-readable medium. Figure 1 makes clear that Herbert's random number generator 18 is distinct from both the processor 16 and any computer-readable medium (i.e., RAM 14, flash memory 15).

Furthermore, the keying material in Herbert is generated for the encryption engine 12 (col. 3, lines 1-3). Encryption keys are used to decrypt pages being paged back into the physically secure environment from external storage (col. 7, lines 9-12). Thus, the encryption keys in Herbert are related to whatever encryption algorithm is used to encrypt outgoing pages. By contrast, Applicant's claim 14 recites "an operating system storage key based on an identity for the operating system". Herbert does not discuss an operating system storage key that is generated based on an identity for the operating system. Herbert's encryption keys are based on an encryption algorithm, not on the identity of an operating system.

For at least the reasons discussed above, it is clear that Herbert does not teach the elements of Applicant's claim 14. Applicant therefore respectfully requests that the §102(b) rejection of claim 14 be withdrawn.

In rejecting **claim 23**, the Office asserts that Herbert teaches a computer-readable medium having computer-executable instructions stored thereon to cause a server computer to perform a method of entering into a secure connection with a client computer, obtaining a session key specific to the secure connection, encrypting data with the session key, and downloading the encrypted data to the client computer. However, as discussed above with respect to claim 1, Herbert teaches a secure method of paging within a computer. Herbert does not discuss downloading data as between a server computer and a client computer. Herbert discloses a "method and system for maintaining integrity and confidentiality of pages paged to an external storage unit from a physically secure environment" (col. 1, lines 58-60). Therefore, among other things, Herbert does not teach a server computer "entering into a secure connection with a client computer" or "downloading the encrypted data to the client computer". For at least these reasons, claim 23 is not anticipated by Herbert, and the §102(b) rejection of claim 23 should be withdrawn.

In rejecting **claim 24**, the Office asserts, among other things, that Herbert teaches computer-executable instructions that cause a client computer to enter into a secure connection with a server computer. However, as discussed above, Herbert teaches maintaining the integrity of pages paged to an external storage unit from a physically secure environment within the context of a virtual memory system in a single computer. Nowhere does Herbert discuss a client computer entering into a secure connection with a server computer. Among other things,

Applicant's claim 24 recites "computer-executable instructions . . . to cause a client computer to perform a method comprising: entering into a secure connection with a server computer". Herbert does not teach the elements of Applicant's claim 24. For at least these reasons, claim 24 is not anticipated by Herbert, and the §102(b) rejection of claim 24 should be withdrawn.

Allowable Subject Matter

Applicant appreciates the allowance of claims 18-22.

Claim Objections

Claims 3-4 and 6-12 are objected to as being dependent upon rejected base claim 1. Claims 15-17 are objected to as being dependent upon rejected base claim 14. Because base claims 1 and 14 are allowable as discussed herein above, dependent claims 3-4, 6-12 and 15-17 are also allowable by virtue of their respective dependencies from base claims 1 and 14.

Conclusion

All pending claims are in condition for allowance. Applicant respectfully requests reconsideration and prompt issuance of the subject application. If any issues remain that prevent issuance of this application, the Examiner is urged to contact the undersigned attorney before issuing a subsequent Action.

Respectfully Submitted,

Dated: 3/4/02

By: _____

Nathan R. Rieth
Reg. No. 44302
(509) 324-9256

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application Serial No. 09/227,368
Filing Date 1/08/99
Inventorship ..... England
Applicant ..... Microsoft Corporation
Group Art Unit 2662
Examiner ..... Jack, Todd M.
Attorney's Docket No. MS1-282USC3
Title: Key-Based Secure Storage

Amended Claims With Markings To Show Changes Made

Claims 1, 2, 8, 11, 12 and 18-22 have been changed by the accompanying Response To Office Action relative to their immediate prior versions. A marked up version of claims 1, 2, 8, 11, 12 and 18-22 is therefore submitted below in accordance with 37 C.F.R. §1.121(c).

1. (Amended) A computerized method for key-based secure storage comprising:

downloading information [content] and an access predicate that specifies requirements for an application to access the information [content];

obtaining a storage key;

encrypting the information [content] using the storage key; and

associating the access predicate with the encrypted information [content].

2. (Amended) The computerized method of claim 1, further comprising:

decrypting the information [content] for access by an application only if the application meets the requirements specified in the access predicate.

8. (Amended) The computerized method of claim 1, wherein the storage key comprises an application storage key and a user storage key to encrypt information [content] containing portion specific to an application and a portion specific to a user, and obtaining the storage key comprises:

generating a seed value for the application;

producing an application hash seed value based on the seed value for the application using an application-specific one-way hash function;

generating an application storage key from the application hash seed value;

generating a seed value for the user;

producing a first user hash seed value based on the seed value for the user using a one-way hash function;

producing a second user hash seed value based on the first user hash seed value and a user identifier using a keyed hash function; and

generating a user storage key from the second user hash seed value.

11. (Amended) The computerized method of claim 9, further comprising:

selecting the key vault from a plurality of key vaults provided by a trusted [digital rights management] operating system.

12. (Amended) The computerized method of claim 9, further comprising:

selecting the key vault designated by a provider of the information [content].

18. (Amended) A computer system comprising:

a processing unit;

a system memory coupled to the processing unit through a system bus;

a computer-readable medium coupled to the processing unit through a system bus; and

a trusted [digital rights management] operating system executed from the computer-readable medium by the processing unit, wherein the trusted [digital rights management] operating system causes the processing unit to encrypt downloaded information [content] using a storage key based on a seed value.

19. (Amended) The computer system of claim 18, wherein the trusted [digital rights management] operating system further causes the processing unit to encrypt an access predicate associated with the downloaded information [content] using an operating system storage key, to encrypt the seed value for the storage key using the operating system storage key, and to associate the encrypted access predicate with the encrypted seed value.

20. (Amended) The computer system of claim 19, wherein the trusted [digital rights management] operating system further causes the processing unit to

validate each application requesting access to the downloaded information [content] using the access predicate, and decrypts the seed value for use by a validated application.

21. (Amended) The computer system of claim 18, wherein the storage key used to encrypt the downloaded information [content] is specific to an application.

22. (Amended) The computer system of claim 18, wherein the storage key used to encrypt the downloaded information [content] is specific to a user.

Respectfully Submitted,

Dated: 3/4/02

By: _____

Nathan R Rieth
Reg. No. 44302
(509) 324-9256; X233